NEW UNITED STATES UTILITY PATENT APPLICATION
under 37 C.F.R. 1.53(b)

Director of the U.S. Patent and Trademark Office
Box Patent Applications
Washington, D.C. 20231

Enclosed herewith is a new patent application and the following papers:

First Named Inventor (or application identifier): Colin S. Cole, Angela Mills, Daniel J. Rogers, and Marcelo R. Uemra

Title of Invention: METHOD FOR REQUEST AND RESPONSE DIRECT DATA TRANSFER AND MANAGEMENT OF CONTENT MANIFESTS

1. ■ Specification 18 pages (including specification, claims, abstract) / 22 claims (4 independent)

2. ■ Unexecuted Declaration/Power of Attorney is:
   - ■ attached in the regular manner. Payment is deferred under 37 C.F.R. § 1.53(f).
   - □ NOT included, but deferred under 37 C.F.R. § 1.53(f).

3. ■ 6 Distinct sheets of ■ Formal □ Informal Drawings

4. □ Preliminary Amendment.

5. □ Information Disclosure Statement
   - □ Form 1449
   - □ A copy of each cited prior art reference

6. □ Assignment with Cover Sheet.

7. ■ Priority is hereby claimed under 35 U.S.C. § 119(e)(1) based upon the following application:

U.S. Provisional Application No.	Date of Filing (day, month, year)	Priority Claimed Under 35 U.S.C. § 119(e)(1)
TBD	25/05/00	Yes

8. □ Priority document(s).

Priority is hereby claimed under 35 U.S.C. § 119 based upon the following application(s):

Country	Application Number	Date of Filing (day, month, year)

9. □ Statement Claiming Small Entity Status.

10. □ Microfiche Computer Program (Appendix).

11. □ Nucleotide and/or Amino Acid Sequence Submission.

NEW UNITED STATES UTILITY PATENT APPLICATION
under 37 C.F.R. 1.53(b)

Atty. Docket No. 3797.86783

□ Computer Readable Copy.

□ Paper Copy (identical to computer copy).

□ Statement verifying identity of above copies.

12. Calculation of Fees:

FEES FOR	EXCESS CLAIMS	FEE	AMOUNT DUE
Basic Filing Fee (37 C.F.R. § 1.16(a))			$690.00
Total Claims in Excess of 20 (37 C.F.R. § 1.16(c))	2	18.00	$36.00
Independent Claims in Excess of 3 (37 C.F.R. § 1.16(b))	1	78.00	$78.00
Multiple Dependent Claims (37 C.F.R. § 1.16(d))	0	260.00	
Subtotal - Filing Fee Due			$804.00
	REDUCE BY (%)		($)
Reduction by 50%, if Small Entity (37 C.F.R. §§ 1.9, 1.27, 1.28)	0.500		
TOTAL FILING FEE DUE			
Assignment Recordation Fee (if applicable) (37 C.F.R. § 1.21(h))	10	40.00	
GRAND TOTAL DUE			**$804.00**

13. PAYMENT is:

□ included in the amount of the GRAND TOTAL by our enclosed check. A general authorization under 37 C.F.R. § 1.25(b), second sentence, is hereby given to credit or debit our Deposit Account No. 19-0733 for the instant filing and for any other fees during the pendency of this application under 37 C.F.R. §§ 1.16, 1.17 and 1.18.

■ not included, but deferred under 37 C.F.R. § 1.53(f).

14. All correspondence for the attached application should be directed to:

Pamela I. Banner
Banner & Witcoff, Ltd.
1001 G Street, N.W.
Washington, D. C. 20001-4597
Telephone: (202) 508-9100
Facsimile: (202) 508-9299

15. Other: ______________________________

Date: 6/29/00

By: Charles Miller 43,805 for
Pamela I. Banner
Reg. No. 33,644

MS Ref. No.: 150453.1
B&W Ref. No.: 3797.86783
Inventor: Daniel Rogers

METHOD FOR REQUEST AND RESPONSE DIRECT DATA TRANSFER AND MANAGEMENT OF CONTENT MANIFESTS

BACKGROUND OF THE INVENTION

This application claims the benefit of U.S. Provisional Application No. 60/------ filed May 25, 2000, the entire disclosure of which is hereby incorporated by reference.

1. Technical Field

The present invention relates to the field of the automated exchange of computerized data. More particularly, the invention provides methods and systems that allow for the automated exchange of information between application programs running on different operating systems.

2. Related Information

Businesses and individuals have become increasing reliant upon the automated exchange of data between computers, computer programs and software applications. Businesses and individuals automatically interact between their own internal line of business, productivity and knowledge management applications, the applications used by their customers and partners, and services provided by their commercial and corporate providers. The use of a variety of different operating systems and software applications by those wishing to automatically exchange data has made the exchange process difficult and expensive.

Conventionally, programmers at two businesses wishing to automatically exchange data would have to agree on how the information would be exchanged. For example, when company A decides to order 10 parts having a part number 3614 for delivery on April 28, 2001, company A would send company B a data stream similar to 10,3614,April 28,2001. The programmers at each company would have to modify application programs at the two locations to include computer code for the formation, transport and protocol used for the exchange of such information. Furthermore, each filed would have to be a predetermined length and transmitted in the correct order.

What is needed is an approach that allows for the efficient automated exchange of data between application programs operating on different operating system platforms.

SUMMARY OF THE INVENTION

The present invention provides a framework that allows for the efficient exchange of data between application programs even when the application programs are operating on different operating system platforms.

The present invention provides a method for exchanging data between a source location and a destination location. The method includes the initial step of generating a data file with a markup language in accordance with a predetermined schema. A software envelope containing the data file is then generated and transmitted to the destination location. At the destination location, an object is created from the data file with a plugin object corresponding to the predetermined schema.

A second envelope maybe automatically be generated from the information contained in the first software envelope. The second envelope may have a destination address matching the source address of the first envelope. Alternatively, the second envelope may have a destination address determined by state information contained in the first envelope.

The software envelope may be transmitted via electronic mail, HTTP or an intermediate server.

The advantages of the present invention may also be provided by a computer-readable medium having stored thereon a particular data structure. The data structure includes: (a) a first field containing address information; (b) a second data field containing the identification of a predetermined schema; and (c) a third data field containing a data file formatted in a markup language in accordance with the schema.

Other features and advantages of the invention will become apparent with reference to the following detailed description and the figures.

BRIEF DESCRIPTION OF THE DRAWINGS

The present invention is illustrated by way of example and not limitation in the accompanying figures in which like reference numerals indicate similar elements and in which:

Figure 1 is a schematic diagram of a conventional general-purpose digital computing environment that can be used to implement various aspects of the invention;

Figure 2 is a block diagram of a configuration used to exchange data;

Figure 3 is a listing of tags that contain envelope and data information;

Figure 4 is a listing of tags that contain delivery information;

Figure 5 is a listing of tags that contain manifest information; and

Figure 6 illustrates a part order schema.

DETAILED DESCRIPTION OF THE INVENTION

Although not required, the invention will be described in the general context of computer-executable instructions, such as program modules, that are executed by a personal computer or a server. Generally, program modules include routines, programs, objects, components, data structures, etc., that perform particular tasks or implement particular abstract data types. Moreover, those skilled in the art will appreciate that the invention may be practiced with other computer system configurations, including hand-held devices, multiprocessor systems, microprocessor-based or programmable consumer electronics, network PCS, minicomputers, mainframe computers, and the like. The invention may also be practiced in distributed computing environments where tasks are performed by remote processing devices that are linked through a communications network. In a distributed computing environment, program modules may be located in both local and remote memory storage devices.

Figure 1 is a schematic diagram of a conventional general-purpose digital computing environment that can be used to implement various aspects of the invention. Computer 100 includes a processing unit 110, a system memory 120 and a system bus 130 that couples various system components including the system memory to the processing unit 110. System bus 130 may be any of several types of bus structures including a memory bus or memory controller, a peripheral bus, and a local bus using any of a variety of bus architectures. System memory 120 includes a read only memory (ROM) 140 and a random access memory (RAM) 150.

A basic input/output system (BIOS) 160 containing the basic routines that help to transfer information between elements within the computer 100, such as during start-up, is stored in ROM 140. Computer 100 also includes a hard disk drive 170 for reading from and writing to a hard disk (not shown), a magnetic disk drive 180 for reading from or writing to a removable magnetic disk 190, and an optical disk drive 191 for reading from or writing to a removable optical disk 192, such as a CD ROM or other optical media. Hard disk drive 170, magnetic disk drive 180, and optical disk drive 191 are respectively connected to the system bus 130 by a hard disk drive interface 192, a magnetic disk drive interface 193, and an optical disk drive interface 194. The drives and their associated computer-readable media provide nonvolatile storage of computer readable instructions, data structures, program modules and other data for personal computer 100. It will be appreciated by those skilled in the art that other types of computer readable media which can store data that is accessible by a computer, such as magnetic cassettes, flash memory cards, digital video disks, Bernoulli cartridges, random access memories (RAMs), read only memories (ROMs), and the like, may also be used in the exemplary operating environment.

A number of program modules can be stored on the hard disk, magnetic disk 190, optical disk 192, ROM 140 or RAM 150, including an operating system 195, one or more application programs 196, other program modules 197, and program data 198. A user can enter commands and information into computer 100 through input devices, such as a keyboard 101 and a pointing device 102. Other input devices (not shown) may include a microphone, joystick, game pad, satellite dish, scanner, or the like. These and other input devices are often connected to the processing unit 110 through a serial port interface 106 that is coupled to the system bus, but may be connected by other interfaces, such as a parallel port, a game port, a universal serial bus (USB) or through a PCI board. A monitor 107 or other type of display device is also connected to system bus 130 via an interface, such as a video adapter 108. In addition to the monitor, personal computers typically include other peripheral output devices (not shown), such as speakers and printers.

Computer 100 can operate in a networked environment using logical connections to one or more remote computers, such as a remote computer 109. Remote computer 109 can be a server, a router, a network PC, a peer device or other common network node, and typically includes many or all of the elements described above relative to computer 100, although only a memory storage device 111 has been illustrated in Figure 1. The logical connections depicted in Figure 1 include a local area network (LAN) 112 and a wide area network (WAN) 113. Such networking environments are commonplace in offices, enterprise-wide computer networks, intranets and the Internet.

When used in a LAN networking environment, computer 100 is connected to local network 112 through a network interface or adapter 114. When used in a WAN networking environment, personal computer 100 typically includes a modem 115 or other means for establishing a communications over wide area network 113, such as the Internet. Modem 115, which may be internal or external, is connected to system bus 130 via serial port interface 106.

In a networked environment, program modules depicted relative to personal computer 100, or portions thereof, may be stored in the remote memory storage device.

It will be appreciated that the network connections shown are exemplary and other ways of establishing a communications link between the computers can be used. The existence of any of various well-known protocols, such as TCP/IP, Ethernet, FTP, HTTP and the like, is presumed, and the system can be operated in a client-server configuration to permit a user to retrieve web pages from a web-based server. Any of various conventional web browsers can be used to display and manipulate data on web pages.

Figure 2 is a functional block diagram of elements that may be used in accordance with the present invention to exchange data from a source location to a destination location. Application 200 may be a computer program located at the source location. An adapter 202 may be an application that creates a software envelope 204 that contains data 206. In this patent the term "envelope" refers to information that defines a delivery convention such as one or more of routing information, return routing information and state management information, much in the same way that a postal system envelope defines a convention where routing and delivery can be accomplished independent of the final purpose, processing, disposition, and information of the contents of the envelope. Of course, this list is not intended to limit or define the exact information of any particular envelope, but simply illustrate the types of information that may be included in an envelope. In an alternative embodiment, application 200 may contain computer code for performing the function of adapter 202. An example of software envelope 204 is described in detail below. A server 208 may handle and route software envelope 204 to the destination location. A single server is shown for illustration purposes only and with the understanding that two or more servers may be connected between the source and destination points as is conventionally done in computer networks such as the Internet. Furthermore, software envelope 204 may be transmitted between application 200 and application 212 via any number of data transmission mechanisms including electronic mail and HTTP. An adapter 210 extracts data 206 from software envelope 204 for further processing by an application 212.

The present invention may use a markup language, such as the extensible markup language (XML) or the standard generalized markup language (SGML) to create and transmit envelopes and data. Figure 3 shows XML tags that may be used to form an envelope that contains data. The <envelope type> tags 302 may be used to identify the type of envelope and an associated namespace 303 so that the envelope can be processed by the server. Namespace 303 may identify element declarations. When using the BizTalk Framework and BizTalk servers, opening tag 302 may be replaced with <biztalk_1 xmlns="urn:biztalk-org:biztalk:biztalk_1">. The information contained within the <header> tags 304 may contain delivery and message content information. Delivery information may be contained within <delivery> tags 306 and message content information may be contained within <manifest> tags 308. Furthermore, data 206 (shown in figure 2) may be contained within <body> tags 310.

Figure 4 shows the type of information that may be included within <delivery> tags 306. The information between <message> tags 402 may contain unique information specific to the particular data being exchanged. For example, an envelope identification code 403 may be contained within <messageID> tags 404. Envelope identification code 403 may be used for logging, tracking, error handling, or other message processing/reprocessing requirements. Application 200 or adapter 202 (shown in figure 2) may generate a unique envelope identification code for each envelope. The <sent> tags 406 may contain a time stamp 405 indicating when the envelope was transmitted from an application. Time stamp 405 may use the ISO 8601 format.

Text 407 describing the contents of the envelope may be contained within <subject> tags 408. For example, figure 4 includes text 407 indicating that the envelope contains data for ordering parts.

The destination location for the envelope may be contained within the <to> tags 410. The information contained within <from> 412 pertains to the source location and may be similar to the information contained within <to> tags 410 and described below. A Universal Resource Identifier (URI) describing the logical address of a destination location may be included within <address> tags 414. The logical address of a source or destination location may be independent of the transport mechanism used to transport the envelope. However, if Application 200 or adapter 202 (shown in figure 2) is already configured with a transport specific URL 413 for the destination system, it can be specified in place of the URI. In one embodiment of the invention, server 208 (shown in figure 2) may be used to resolve a destination location URI into the appropriate transport-specific address when provided with a logical identifier of the destination location.

State information may be contained within <state> tags 416. State information may be used to correlate individual messages with specific exchanges and processes and may include interchange, handle, and state identifiers. For example, state information may instruct the recipient of an envelope to send replies to an address that is different from the source location.

The present invention allows the recipient of an envelope containing data to conveniently reply to the sender. In particular, the information making up the envelope can be used to automatically create a new reply envelope. In one embodiment of the invention, application 212 or adapter 210 (shown in figure 2) creates a new envelope with the source and destination address information reversed. Furthermore, state information may be used to indicate that a response should be sent to an address other than the source address. For example, state information attached to a purchase order may indicate that an order acknowledgement reply should be sent to a second address while a shipping notice reply should be sent to a third address.

Figure 5 shows the type of information that may be contained within the <manifest> tags 308. Such information may indicate the type of information contained in the software envelope. The identification of each data file 206, shown in figure 2 and described in detail below, is found between a pair of <document> tags 502. Figure 5 shows a single document 501 for illustration purposes only. When two or more documents are sent in the same envelope, they may be listed between the <manifest> tags 308 in the same order that they appear between <body> tags 310 to facilitate the processing of the information. The name of the data file corresponding to element 206, shown in figure 2, may be contained within <name> tags 504. A textual description of the data file may be contained within <description> tags 508. Comments that may be useful or requested by the source location, such as keywords that facilitate searching, may be include between <description> tags 508.

The present invention also allows for the attachment of one or more files to the software envelope by describing the attachments between a pair of <attachment> tags 510 for each attached file. The name of an attached file may be included between <filename> tags 512 and a textual description of the attached file may be contained within <description> tags 514. The type of file may be included between <type> tags 516. There are many situations in which a user may want to attach a file to the software envelope. For example, a user may want to attach prices found in an advertisement distributed by a supplier to an envelope containing a part order request. The user can scan the advertisement and create an image in JPG format. Figure 5 shows what the manifest format may look like when a user attaches a JPG image of an advertisement to a part order request. When more than one file is attached to a software envelope, each attachment may be identified by a unique index number placed between <index> tags (not shown).

Data file 206 (shown in figure 2) may be contained between <body> tags. Data file 206 is preferably marked up with a markup language that matches the markup used for software envelope 204. The format of the marked up file will be in accordance with a schema agreed upon by the source and destination locations. Figure 6 shows an example of a marked up data file. Opening schema tag 602 identifies the schema as "part_order_3." A namespace reference 603 may be included to distinguish namespace 603 associated with the schema from namespace 303 associated with opening element type tag 302. A variety of information and corresponding tags make up the document 604, which corresponds to data 206 shown in figure 2. As an example, figure 6 shows a document 604 used for ordering parts. The identification of the schema allows application 212 or adapter 210 to use an appropriate plugin to parse and extract the data from the data file. A plugin may implement the design pattern by which object constructors are linked with Document Object Model (DOM) nodes that are contained within the software envelope. For example, a plugin constructed in accordance with schema part_order_3 would know that the part number corresponding to the part being ordered would be found between the <partNo> tags. Furthermore, a schema may indicate that certain information is optional. For example, the description information found between the <description> tags shown in figure 6 may be optional. The corresponding plugin may extract optional information if it is present.

The appropriate plugin may depend upon the operating system utilized by the destination location. For example, a first plugin may extract the information contained in the data file shown in figure 6 and rehydrate the information into an object recognized by the first operating system. Furthermore, a second plugin may be used to extract and rehydrate the same information into an object recognized by a second operating system. In one embodiment of the invention, a plugin or parser may be attached to the software envelope.

While the present invention has been described in connection with the illustrated embodiments, it will be appreciated and understood that modifications may be made without departing from the true spirit and scope of the invention.

MS Ref. No.: 150453.1
B&W Ref. No.: 3797.86783
Inventor: Daniel Rogers

What is claimed is:

1. A method for exchanging data between a source location and a destination location comprising the steps of:

generating a data file with a markup language in accordance with a predetermined schema;

generating a first software envelope containing the data file;

transmitting the software envelope to the destination location; and

creating an object from the data file with a plugin object corresponding to the predetermined schema.

2. The method of claim 2, further including the step of:

automatically generating a second software envelope from the information contained in the first software envelope.

3. The method of claim 2, wherein the first software envelope contains destination and source address information and

wherein the step of automatically generating a second envelope includes generating a second envelope having a destination address matching the source address of the first envelope.

4. The method of claim 2, wherein the first software envelope contains state information and

wherein the step of automatically generating a second envelope includes generating a second envelope having a destination address determined by the state information.

5. The method of claim 1, wherein the markup language comprises extensible markup language (XML).

6. The method of claim 1, wherein the markup language comprises standard generalized markup language (SGML).

7. The method of claim 1, wherein the step of transmitting comprises transmitting the software envelope via electronic mail.

8. The method of claim 1, wherein the step of transmitting comprises transmitting the software envelope via HTTP.

9. The method of claim 1, wherein the step of transmitting comprises transmitting the software envelope via an intermediate server.

10. A computer readable medium having computer-executable instructions for performing the steps recited in claim 1.

11. A computer-readable medium having stored thereon a data structure comprising:

(a) a data field containing address information;

(b) a data field containing the identification of a predetermined schema; and

(c) a data field containing a data file formatted in a markup language in accordance with the schema.

12. The computer readable medium of claim 11, further including:

(d) a data field containing manifest information corresponding to the information contained in the data file data field.

13. The computer readable medium of claim 11, further including:

(d) a data field containing state information.

14. The computer readable medium of claim 13, wherein the state information contains address information.

15. The computer readable medium of claim 11, wherein the address information contains an address for replying to a message.

16. A method for creating data at a source location to transmit to a destination location comprising the steps of:

generating a data file with a markup language in accordance with a predetermined schema;

identifying a plugin object that creates an object from the data file;

generating a software envelope containing the data file; and

transmitting the software envelope to the destination location.

17. The method of claim 16, wherein the step of generating a software envelope includes generating a software envelope containing the data file and the plugin object.

18. The method of claim 16, wherein the markup language comprises extensible markup language (XML).

19. The method of claim 16, wherein the markup language comprises standard generalized markup language (SGML).

20. A method for extracting data from a file transmitted from a source location comprising the steps of:

receiving a software envelope containing a data file marked up with a markup language in accordance with a predetermined schema; and

creating an object from the data file with a plugin object corresponding to the predetermined schema.

21. The method of claim 20, wherein the markup language comprises extensible markup language (XML).

22. The method of claim 20, wherein the markup language comprises standard generalized markup language (SGML).

MS Ref. No.: 150453.1
B&W Ref. No.: 3797.86783
Inventor: Daniel Rogers

ABSTRACT OF THE INVENTION

A method for automatically exchanging computer data is disclosed. The method includes the step of first generating a data file with a markup language in accordance with a predetermined schema. The schema is agreed upon by the source and destination location. Next, a software envelope containing the data file is generated. The software envelope is transmitted to the destination location. At the destination location, an object is created from the data file with a plugin object corresponding to the predetermined schema.

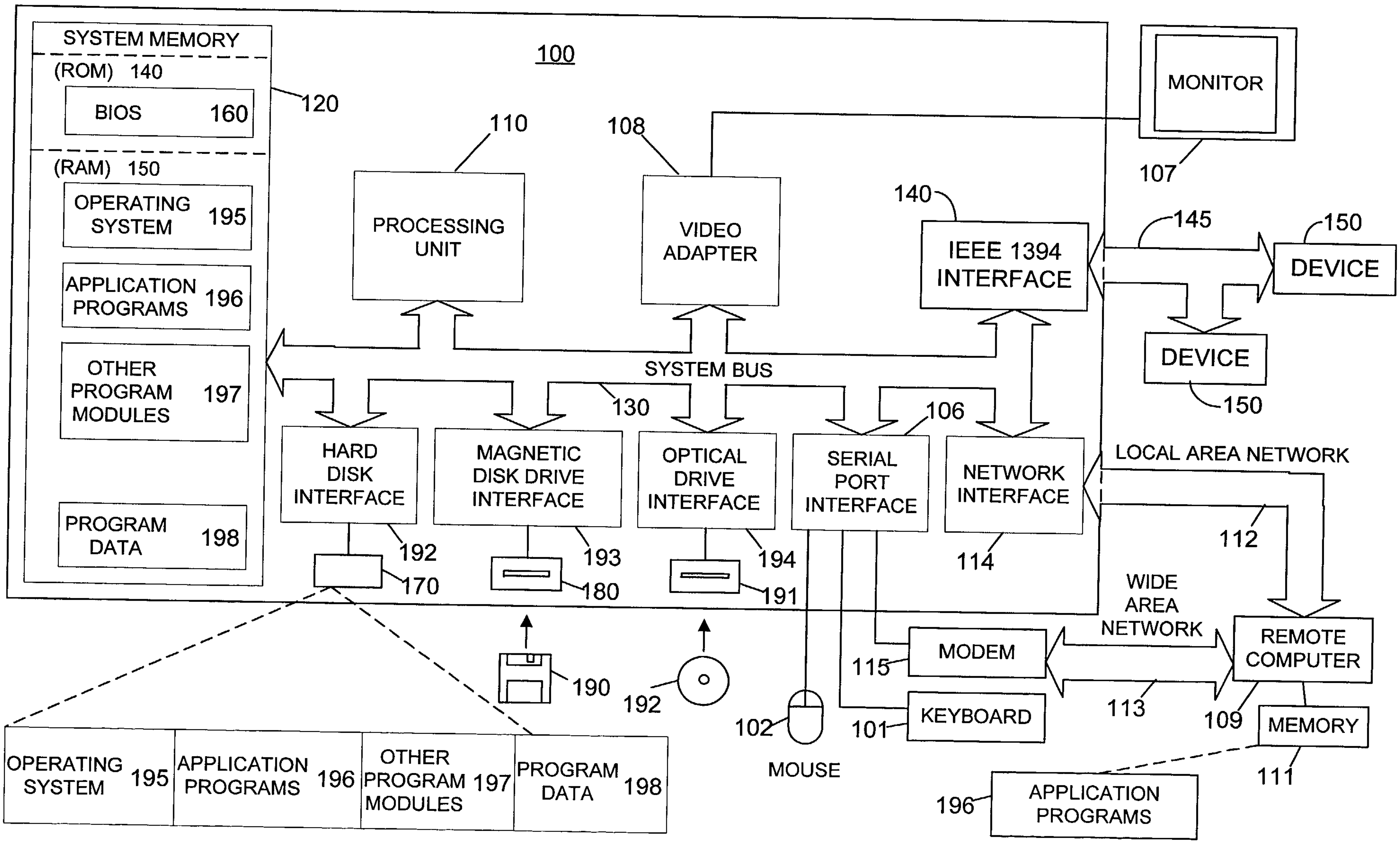

SYSTEM MEMORY
(ROM) 140
BIOS 160
(RAM) 150
OPERATING SYSTEM 195
APPLICATION PROGRAMS 196
OTHER PROGRAM MODULES 197
PROGRAM DATA 198
120
100
110
PROCESSING UNIT
108
VIDEO ADAPTER
MONITOR
107
140
IEEE 1394 INTERFACE
145
150
DEVICE
DEVICE
150
SYSTEM BUS
130
106
HARD DISK INTERFACE
MAGNETIC DISK DRIVE INTERFACE
OPTICAL DRIVE INTERFACE
SERIAL PORT INTERFACE
NETWORK INTERFACE
LOCAL AREA NETWORK
192
193
194
114
112
170
180
191
WIDE AREA NETWORK
REMOTE COMPUTER
MODEM
115
190
192
102
MOUSE
101
KEYBOARD
113
109
MEMORY
111
196
APPLICATION PROGRAMS
OPERATING SYSTEM 195
APPLICATION PROGRAMS 196
OTHER PROGRAM MODULES 197
PROGRAM DATA 198

FIG. 1

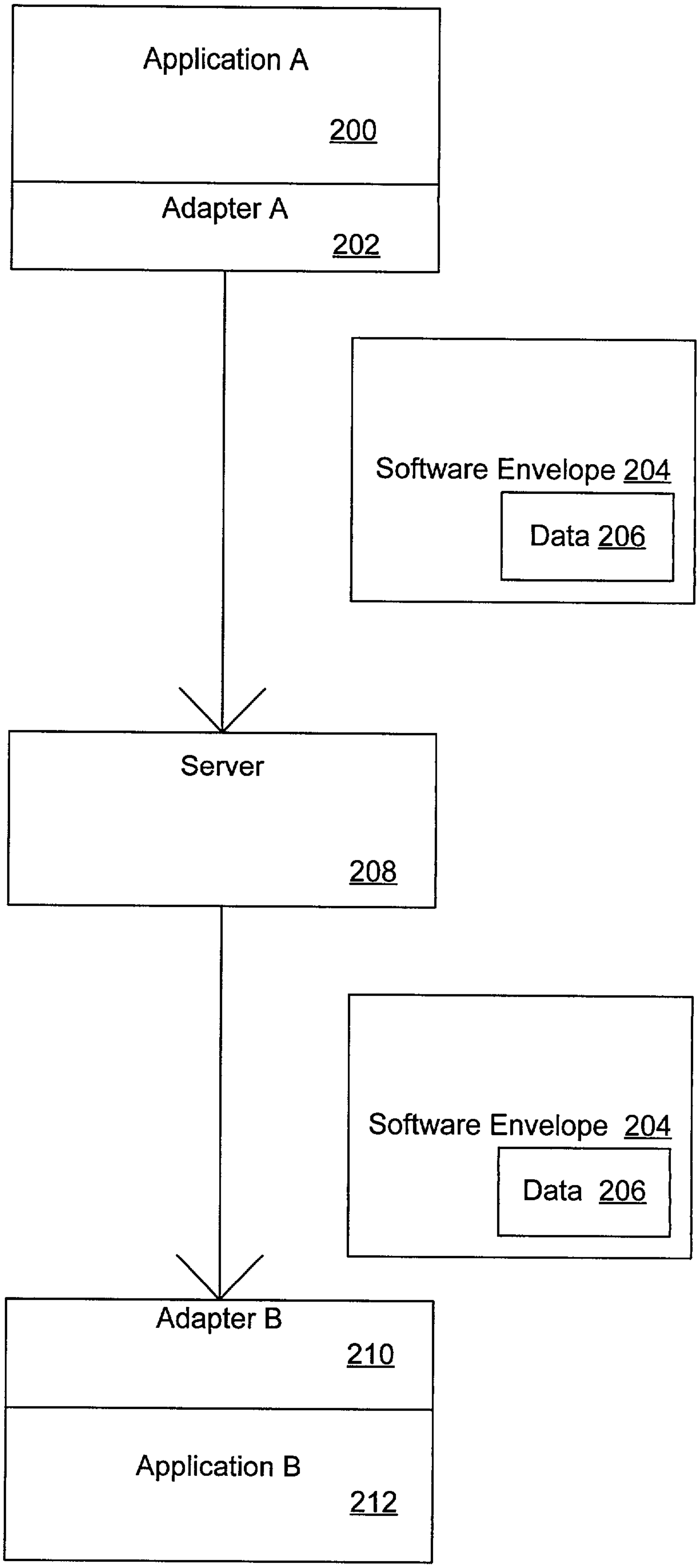
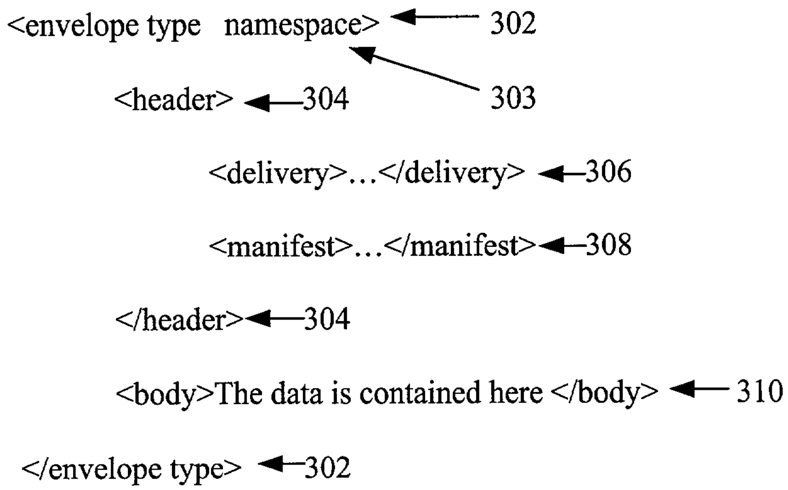
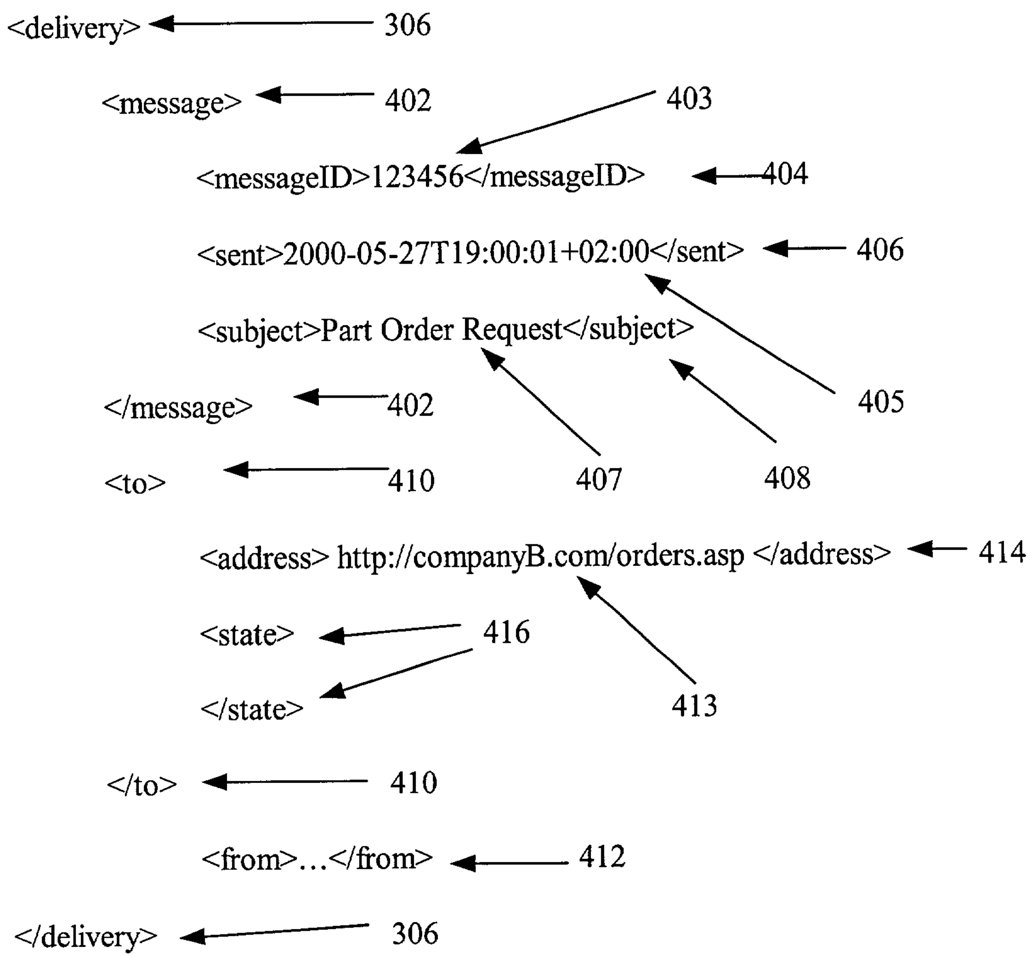

Application A
200
Adapter A
202
Software Envelope 204
Data 206
Server
208
Software Envelope 204
Data 206
Adapter B
210
Application B
212

FIG. 2



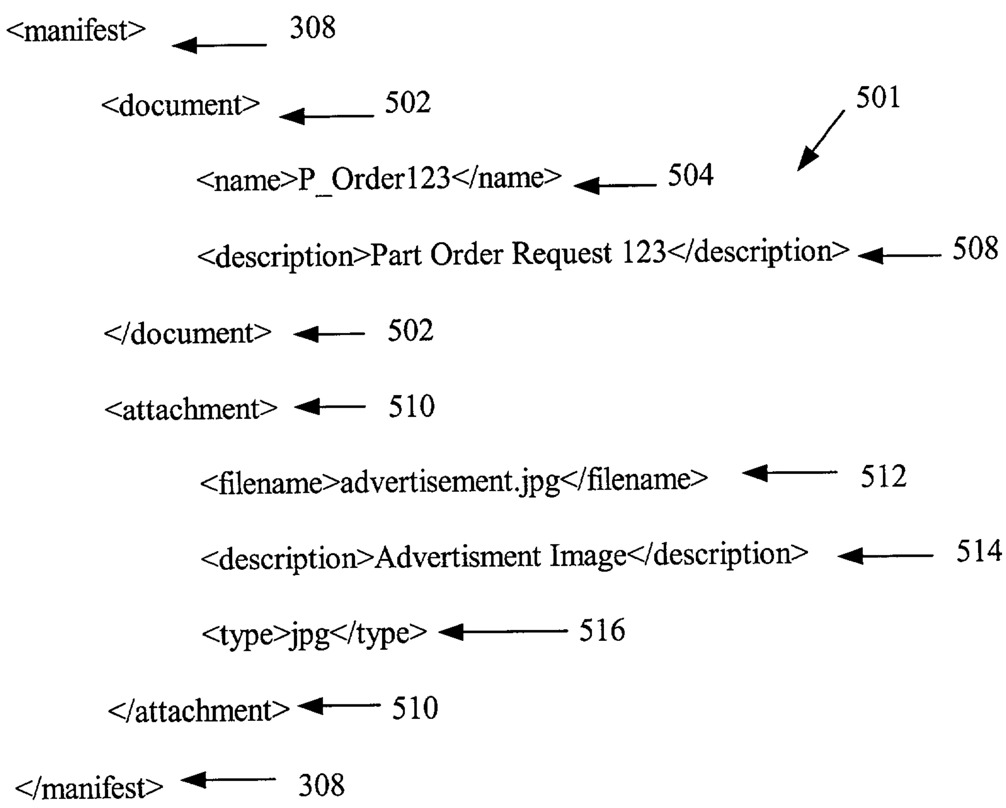
```
<envelope type   namespace>
        <header>
                <delivery>...</delivery>
                <manifest>...</manifest>
        </header>
        <body>The data is contained here </body>
 </envelope type>
```

FIG. 3



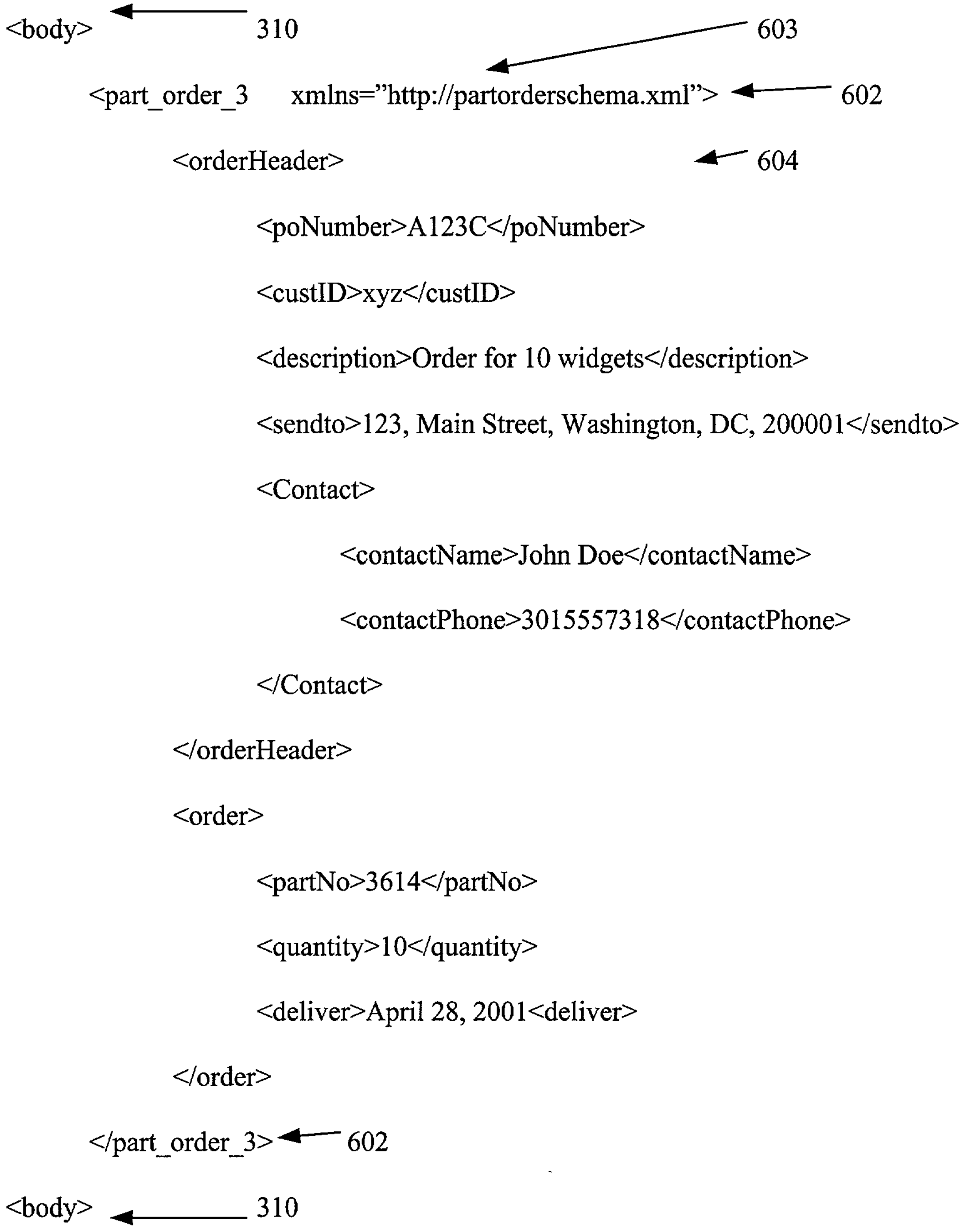
```
<delivery>
        <message>
                <messageID>123456</messageID>
                <sent>2000-05-27T19:00:01+02:00</sent>
                <subject>Part Order Request</subject>
        </message>
        <to>
                <address> http://companyB.com/orders.asp </address>
                <state>
                </state>
        </to>
                <from>…</from>
</delivery>
```

FIG. 4



```
<manifest>
        <document>
                <name>P_Order123</name>
                <description>Part Order Request 123</description>
        </document>
        <attachment>
                <filename>advertisement.jpg</filename>
                <description>Advertisment Image</description>
                <type>jpg</type>
        </attachment>
</manifest>
```

FIG. 5



```
<body>
    <part_order_3    xmlns="http://partorderschema.xml">
        <orderHeader>
            <poNumber>A123C</poNumber>
            <custID>xyz</custID>
            <description>Order for 10 widgets</description>
            <sendto>123, Main Street, Washington, DC, 200001</sendto>
            <Contact>
                <contactName>John Doe</contactName>
                <contactPhone>3015557318</contactPhone>
            </Contact>
        </orderHeader>
        <order>
            <partNo>3614</partNo>
            <quantity>10</quantity>
            <deliver>April 28, 2001<deliver>
        </order>
    </part_order_3>
<body>
```

FIG. 6

JOINT DECLARATION FOR PATENT APPLICATION

As the below named inventors, we hereby declare that:

Our residences, post office addresses and citizenships are as stated below next to our names;

We believe we are the original, first and joint inventors of the subject matter which is claimed and for which a patent is sought on the invention entitled Method for Request and Response Direct Data Transfer and Management of Content Manifests, the specification of which

■ is attached hereto.

□ was filed on __________ as Application Serial Number ____ and was amended on __________ (if applicable).

□ was filed under the Patent Cooperation Treaty (PCT) and accorded International Application No. ___________, filed __________, and amended on __________________ (if any).

We hereby state that we have reviewed and understand the contents of the above identified specification, including the claims, as amended by any amendment referred to above.

We hereby acknowledge the duty to disclose information which is material to patentability in accordance with Title 37, Code of Federal Regulations, §1.56(a).

Prior Foreign Application(s)

We hereby claim foreign priority benefits under Title 35, United States Code, §119 of any foreign application(s) for patent or inventor's certificate listed below and have also identified below any foreign application(s) for patent or inventor's certificate having a filing date before that of the application on which priority is claimed:

Country	Application No.	Date of Filing (day month year)	Date of Issue (day month year)	Priority Claimed Under 35 U.S.C. §119

Prior United States Provisional Application(s)

We hereby claim priority benefits under Title 35, United States Code, §119(e)(1) of any U.S. provisional application listed below:

U.S. Provisional Application No.	Date of Filing (day month year)	Priority Claimed Under 35 U.S.C. §119(e)(1)
TBD	25/05/00	Yes

Prior United States Application(s)

We hereby claim the benefit under Title 35, United States Code, §120 of any United States application(s) listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application in the manner provided by the first paragraph of Title 35, United States Code, §112, we acknowledge the duty to disclose material information as defined in Title 37, Code of Federal Regulations, §1.56(a) which occurred between the filing date of the prior application and the national or PCT international filing date of this application:

Application Serial No.	Date of Filing (Day, Month, Year)	Status — Patented, Pending, Abandoned

Power of Attorney

And we hereby appoint, both jointly and severally, as our attorneys with full power of substitution and revocation, to prosecute this application and to transact all business in the Patent and Trademark Office connected herewith the following at attorneys and agents, their registration numbers being listed after their names:

ALTHERR, Robert F.	31,810	HOSCHEIT, Dale H.	19,090	NIEGOWSKI, James A.	28,331
BANNER, Donald W.	17,037	HYMEL, Lin J.	45,414	PATEL, Binal J.	42,065
BANNER, Mark T.	29,888	IWANICKI, John P.	34,628	PATHAK, Ajay S.	38,266
BANNER, Pamela I.	33,644	JACKSON, Thomas H.	29,808	PAYNE, Stephen S.	35,316
BECKER, Matthew P.	45,824	KAGAN, Sarah A.	32,141	PETERSON, Thomas L.	30,969
BECKETT, William W.	18,262	KATZ, Robert S.	36,402	POTENZA, Joseph M.	28,175
BERGHAMMER, Joseph J.	46,057	KLEIN, William J.	43,719	PRATT, Thomas K.	37,210
BODNER, Jordan	42,338	KRAUSE, Joseph P.	32,578	RENK, Christopher J.	33,761
BUROW, Scott A.	42,373	LINEK, Ernest V.	29,822	RESIS, Robert H.	32,168
CALLAHAN, James V.	20,095	MALONE, Dale A.	32,155	RIVARD, Paul M.	43,446
CHANG, Steve S	42,402	MANNAVA, Ashok K.	45,301	ROBINSON, Douglas W.	32,751
COHAN, Gregory J.	40,959	MAPLE, Marie-Claire B.	37,588	SCHAD, Steven P.	32,550
COOPERMAN, Marc S.	34,143	MAY, Steven A.	44,912	SHIFLEY, Charles W.	28,042
CURTIN, Joseph P.	34,571	McDERMOTT, Peter D.	29,411	SKERPON, Joseph M.	29,864
DeMOOR, Laura J.	39,654	McKEE, Christopher L.	32,384	STOCKLEY, D. J.	34,257
EVANS, Thomas L.	35,805	McKIE, Edward F.	17,335	VAN ES, J. Pieter	37,746
FEDOROCHKO, Gary D.	35,509	MEDLOCK, Nina L.	29,673	WITCOFF, Sheldon W.	17,399
FISHER, Daniel E.	34,162	MEECE, Timothy C.	38,553	WOLFFE, Franklin D.	19,724
FISHER, William J.	32,133	MEEKER, Frederic M.	35,282	WOLFFE, Susan A.	33,568
GLEMBOCKI, Christopher R.	38,800	MILLER, Charles L.	43,805	WRIGHT, Bradley C.	38,061
HANLON, Brian E.	40,449	MITRIUS, Janice V.	43,808	SAKO, Katie E.	32,628
HEMMENDINGER, Lisa M.	42,653	MORENO, Christopher P.	38,566	CROUSE, Daniel D.	32,022
HONG, Patricia E.	34,373	NELSON, Jon O.	24,566		

All correspondence and telephone communications should be addressed to:

Pamela I. Banner
Banner & Witcoff, Ltd.
1001 G Street, N.W., 11th Floor
Washington, D.C. 20001-4597

Customer Number: 22907
Tel: (202) 508-9100
Fax: (202) 508-9299

We hereby declare that all statements made herein of our own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issuing thereon.

Signature____________________ Date____________

Full Name of First Inventor Cole (Family Name) Colin (First Given Name) S. (Second Given Name)

Residence Glenview, IL 60052 Citizenship USA

Post Office Address 214 Nora Avenue, Glenview, IL 60052

Signature____________________ Date__________

Full Name of Second Inventor Mills Angela

Family Name First Given Name Second Given Name

Residence Redmond, WA 98052 Citizenship USA

Post Office Address 18639 NE 55th Way, Redmond, WA 98052

Signature____________________ Date__________

Full Name of Third Inventor Rogers Daniel J.

Family Name First Given Name Second Given Name

Residence Redmond, WA 98052 Citizenship USA

Post Office Address 2122 177th Place, N.E., Redmond, WA 98052

Signature____________________ Date__________

Full Name of Fourth Inventor Uemra Marcelo R.

Family Name First Given Name Second Given Name

Residence Redmond, WA 98052 Citizenship USA

Post Office Address 8702 164th Avenue, N.E., Apt. B202, Redmond, WA 98052